October 21st, 2019

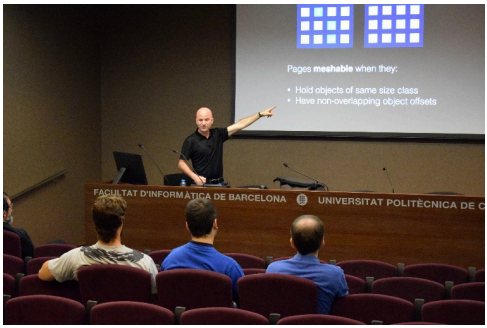
Mesh: Automatically Compacting Memory for C/C++ Applications

Emery Berger

Professor in the College of Information and Computer Sciences at the University of Massachusetts Amherst

Abstract

Programs written in C and C++ — and languages implemented in C, like Python and Ruby — can suffer from serious memory fragmentation, leading to low utilization of memory, degraded performance, and application failure due to memory exhaustion. This talk introduces Mesh, a plug-in replacement for malloc that, for the first time, eliminates fragmentation in unmodified applications through compaction. A key challenge is that, unlike in garbage-collected environments, the addresses of allocated objects in C/C++ are directly exposed to programmers, and applications may do things like stash addresses in integers, and store flags in the low bits of aligned addresses. This hostile environment makes it impossible to safely relocate objects, as the runtime cannot precisely locate and update pointers.



Mesh combines novel randomized algorithms with widely-supported virtual memory operations to provably reduce fragmentation, breaking long-established worst-case bounds on memory efficiency with high probability. Mesh generally matches the runtime performance of state-of-the art memory allocators while reducing memory consumption and eliminating pathological cases; in particular, Mesh reduces the memory of con-

sumption of Firefox by 16% and Redis by 39%. There are efforts underway to incorporate Mesh's approach to eliminate fragmentation into existing allocators like tcmalloc and jemalloc; Mesh itself is available at https://github.com/plasma-umass/Mesh, and it can be used just by setting an environment variable.

Short bio

Emery Berger is a Professor in the College of Information and Computer Sciences at the University of Massachusetts Amherst, the flagship campus of the UMass system, and has been a Visiting Scientist at Microsoft Research and the Polytechnic University of Catalonia (BarcelonaTech). Professor Berger's research spans programming languages, runtime systems, and operating systems, with a particular focus on systems that transparently improve reliability, security, and performance. He is the creator of a number of influential software systems including Hoard, a fast and scalable memory manager that accelerates multithreaded applications (used by companies including British Telecom, Cisco, Crédit Suisse, Reuters, Royal Bank of Canada, SAP, and Tata, and on which the Mac OS X memory manager is based); DieHard, an error-avoiding memory manager that directly influenced the design of the Windows 7 Fault-Tolerant Heap; and DieHarder, a secure memory manager that was an inspiration for hardening changes made to the Windows 8 heap. His honors include a Microsoft Research Fellowship, an NSF CAREER Award, a Lilly Teaching Fellowship, Most Influential Paper Awards at ASPLOS, OOPSLA, and PLDI a Google Research Award, and a Microsoft SEIF Award; he is a Distinguished Member of the ACM. Professor Berger is currently serving a second term as an elected member-at-large of the SIGPLAN Executive Committee. He served as Program Chair for PLDI 2016 and spent a decade as Associate Editor of the ACM Transactions on Programming Languages and Systems.